Rev.Acad.Canar.Cienc., VIII (Núm. 1), 135-142 (1996)

El algoritmo de Disección Unidireccional para mallas regulares

P.R. Almeida Benítez *
U. Las Palmas de Gran Canaria

J.R. Franco Brañas †
U. La Laguna

Resumen. En este artículo se considera el algoritmo de Disección Unidireccional para resolver sistemas lineales de ecuaciones con grafo en forma de malla. Este algoritmo fue diseñado originalmente por Alan George para resolver problemas de aplicaciones de Elementos Finitos. Aquí se compara el modo habitual de reordenar los nodos de los bloques con el reordenamiento obtenido al aplicar el algoritmo de Grado Mínimo a cada bloque. La reducción en el efecto $fill-in$ es mayor que al aplicar los algoritmos de Disección Anidada, Cuthill-McKee o Grado Mínimo.

Palabras Clave. Sistemas sparse, eliminación gaussiana, mallas de Elementos Finitos.

1 Introducción.

Sea A una matriz simétrica definida positiva $n \times n$. Se quiere resolver el sistema de ecuaciones lineales

$$Ax = b$$

donde x y b son vectores columna de orden n y A es una matriz no singular de orden n. Si A es una matriz simétrica y definida positiva, entonces la factorización de *Cholesky* es el mejor método para obtener la descomposición triangular

$$A = LL^t$$

donde L es una matriz triangular inferior. Finalmente, se resuelven los sistemas triangulares $Ly = b$ y $L^t x = y$.

Si A es densa, el número de multiplicaciones/divisiones necesarias para la factorización de *Cholesky* es $n^3/6 + O(n^2)$. Si A es *sparse*, será posible ahorrar tiempo y almacenamiento manipulando los ceros, y será muy ventajoso reducir el efecto $fill-in$ en la factorización de A.

Existen diferentes algoritmos que intentan minimizar el efecto $fill-in$ durante la factorización. Unos lo consiguen mediante una minimización local de

*Departamento of Matemáticas
†Departamento de Análisis Matemático

dicho efecto $fill - in$. Son las estrategias de Markowitz [15], Grado Mínimo [4], etc. Algunos intentan reducir el ancho de banda: Cuthill-McKee [2], King [12], etc. Finalmente, otros intentan dividir el grafo de la matriz A en bloques sin $fill - in$ entre los diferentes bloques: Disección Unidireccional [8], Disección Anidada [4], Go-Away [1], etc.

En el presente artículo se considera el algoritmo de Disección Unidireccional para resolver sistemas de ecuaciones lineales con grafo en forma de malla, comparando el modo habitual de reordenar los bloques con el reordenamiento obtenido al aplicar el algoritmo de Grado Mínimo a cada bloque.

Las líneas generales de este artículo fueron expuestas en el XIV CEDYA - IV Congreso de Matemática Aplicada, celebrado en Vic del 18 al 22 de Septiembre de 1995.

2 El algoritmo de Disección Unidireccional

El algoritmo conocido como Disección Unidireccional fue diseñado por Alan George [7] para resolver problemas que aparecen en aplicaciones de Elementos Finitos. La ventaja del algoritmo es que la demanda de almacenamiento es generalmente menor que en otros algoritmos. Se verá que, con una adecuada renumeración de los nodos de los bloques, este algoritmo permite obtener una reducción sustancial del efecto $fill - in$ con respecto a los algoritmos antes mencionados.

La idea principal del algoritmo consiste en dividir el grafo o malla retirando de él líneas verticales de nodos (llamadas *separadores*) que dividen el grafo en bloques de aproximadamente el mismo tamaño. El método se basa en el paradigma "divide y vencerás". Por ejemplo, si el rectángulo de la figura 1 representa el grafo de la matriz de un sistema $Ax = b$ y se toman tres separadores S_1, S_2 y S_3, el grafo queda dividido en cuatro bloques independientes: B_1, B_2, B_3 y B_4, que se renumeran en primer lugar, comenzando por la última fila de B_1 de izquierda a derecha.

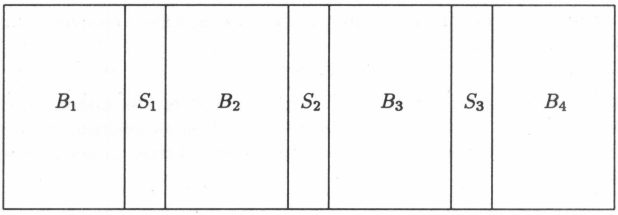

Figura 1

Finalmente, se renumeran los separadores S_1, S_2 y S_3, de abajo arriba. La matriz reordenada A tendrá el perfil de la zona cuadriculada de la figura 2.

Así, los posibles $fill - ins$ occuren en el interior de esas zonas y las zonas punteadas. Por otra parte, dichas zonas no son densas sino con perfil en banda.

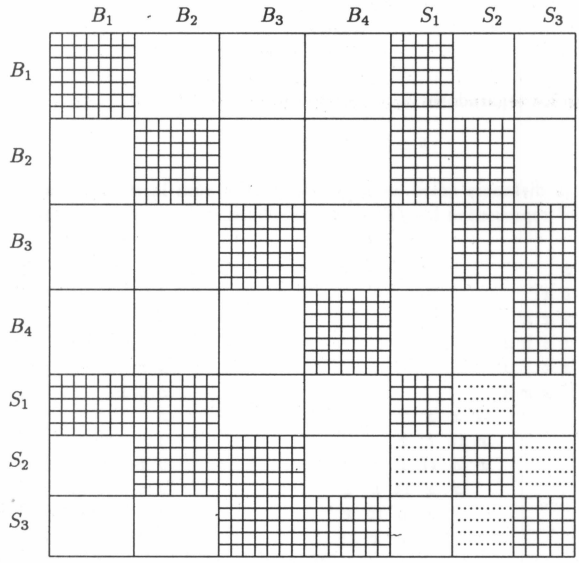

Figura 2

Se considera una malla con m filas y l columnas (con $N = mxl$ nodos) tal que $m \leq l$. Si un elemento de A es distinto de cero, $a_{ij} \neq 0$, entonces los nodos i y j están conectados en las malla mediante una arista. Si se retiran σ columnas de la malla, ésta queda dividida en $\sigma + 1$ bloques independientes ($\sigma \in N, 1 \leq \sigma$). La distancia δ entre separadores será:

$$\delta = (l - \sigma)/(\sigma + 1)$$

y la matriz queda dividida en q^2 submatrices, donde $q = 2\sigma + 1$.

Ejemplo. Sea la malla de la figura 3 (m=3, l=8):

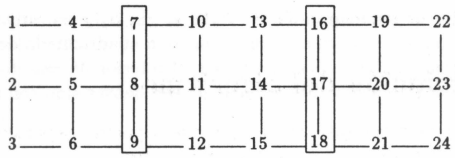

Figura 3

Se toman los separadores S_1 y S_2. Como $\sigma = 2$:

$$\delta = (8-2)/(2+1) = 2 \ .$$

donde δ es la distancia entre los separadores. He aquí la matriz reordenada, donde los "o" representan los $fill-in$ producidos en la eliminación gaussiana:

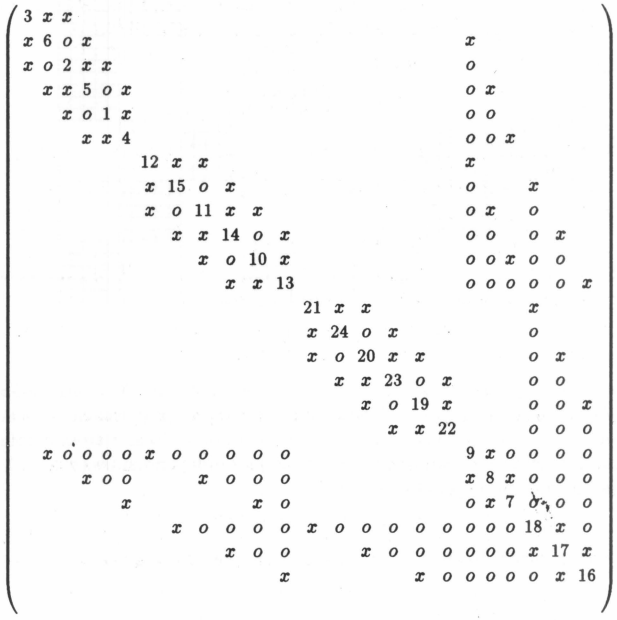

Figura 4

Como se puede apreciar, el factor L de la matriz ha sufrido 47 $fill - ins$.

3 Descripción del algoritmo

El algoritmo intenta encontrar un conjunto de separadores paralelos que dividan el grafo en un conjunto de bloques independientes.

Sea un grafo conexo $G = (X, E)$. Un subconjunto de nodos $Y \subset X$ es un *separador* del grafo conexo G si la sección $G(X - Y)$ es no conexa. Sea $N = |X| = mxl$. Entonces las etapas del algoritmo serán:

Etapa 1 (Generar una estructura de niveles) Encontrar un nodo pseudo-periférico x y generar la estructura de niveles enraizada en x:

$$L(x) = \{L_o, L_1, ..., L_l\}$$

Etapa 2 (Estimar δ) Calcular $\delta = \sqrt{\frac{3x+13}{2}}$. Este valor de δ fue obtenido por Alan George [8] basándose en experimentos numéricos y análisis de mallas regulares al objeto de minimizar los costes de almacenamiento.

Etapa 3 (Encontrar separadores) Sea

$$j = [i\delta + 0.5], \; i = 1, 2, ...$$

donde [ ] es la parte entera de un número, y $S = \emptyset$. Mientras $j < l$ hacer:

3a) Elegir $S_i = \{x \in L_j / Ady(x) \cap L_{j+1} \neq \emptyset\}$.

3b) $S \leftarrow S \cup S_i$.

3c) $S \leftarrow i + 1$ y $j = [i\delta + 0.5]$.

Etapa 4 (Definir los bloques) Sean B_k, $k = 1, 2, ..., p - 1$, las componentes conexas de la sección de grafo $G(X - S)$. Sea $B_p = S$.

Etapa 5 (Numeración de los bloques) Se numera cada bloque B_k, $k = 1, 2, ..., p$, consecutivamente como se indica en la figura 1.

4 Renumeración interna de los bloques

Los nodos de cada bloque han sido renumerados fila por fila como se indica en la figura 1. Desde luego, el efecto $fill - in$ podría reducirse todavía más si se renumeran los nodos atendiendo a su grado (número de conexiones en el grafo) usando el algoritmo de Grado Mínimo. Se modificará la etapa 5 en el algoritmo previo. Sea $P = \{B_1, B_2, ..., B_p\}$. Para cada bloque $B_k, 1 \leq k \leq p - 1$, de grafo asociado $G_o = (Z, E)$ en P, el algoritmo básico será el siguiente:

Etapa 5a (Inicialización) $i \leftarrow 1$.

Etapa 5b (Selección del nodo de grado mínimo) En el grafo de eliminación G_{i-1} se elige un nodo x de grado mínimo.

Etapa 5c (Transformación del grafo) Se forma el nuevo grafo de eliminación G_i eliminando el nodo x en G_{i-1}.

Etapa 5d (Bucle/Parada) $i \leftarrow i+1$. Si $i > |Z|$, stop. en otro caso, volver a etapa 5b.

En la malla anterior resulta el ordenamiento siguiente:

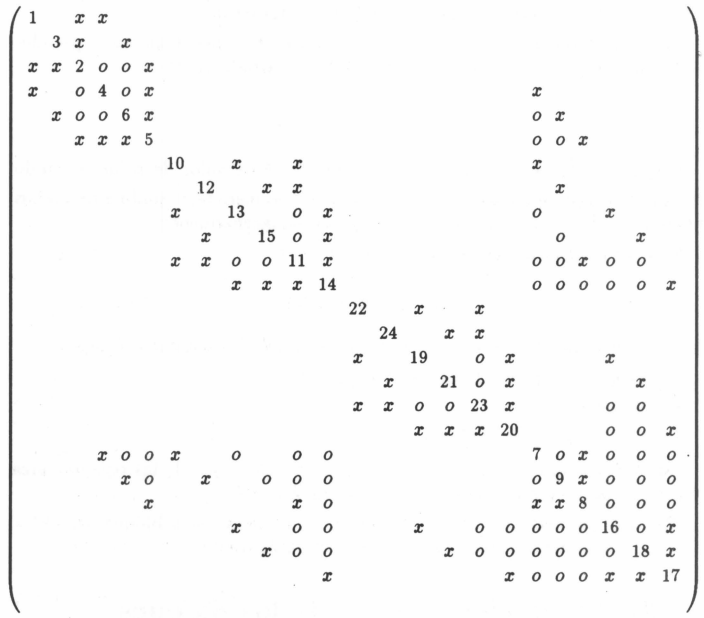

Figura 5

Se observa que el número de $fill-ins$ en el factor triangular L se ha reducido de 47 a 36.

Se va a hacer una tabla resumen del efecto $fill-in$ para distintos algoritmos, Grado Mínimo (GM), Disección Unidireccional (DU), Disección Unidireccional con reordenamiento interno de los bloques (DU(Re)), Disección Anidada (DA),

Cuthill-McKee (CM) e Inverso de Cuthill-McKee (ICM):

Dimension	Natural	GM	DU	DU(Re)	DA	CM	ICM
3x5	28	23	20	13	16	15	14
4x4	27	26	24	14	18	22	22
5x5	64	81	59	48	51	50	49
6x6	125	191	117	94	98	95	95
7x7	210	356	198	152	159	161	160
10x10	619	1181	579	398	426	469	469
20x20	3499	7506	3344	1998	2226	2735	2734

Tabla 1

5 Conclusiones

Se ha visto que un reordenamiento interno de los bloques en el algoritmo de Disección Unidireccional redunda en una disminución del efecto $fill - in$. Esta reducción es mayor que al aplicar los algoritmos de Cuthill-McKee, Inverso de Cuthill-McKee o Disección Anidada a la malla. Desde luego, el Algoritmo de Grado Mínimo produce un mayor $fill - in$ y no es un buen algoritmo para este tipo de mallas.

Por otra parte, suele ocurrir que aparece más de un nodo de grado mínimo en la etapa 2 del algoritmo. Se ha resuelto el problema renumerando los nodos en el orden generado por el algoritmo de Disección Unidireccional, v.g., de un modo prácticamente arbitrario. Desde luego, la estrategia a seguir para resolver los "empates" no es una cuestión trivial en muchos casos. Duff et al. [3] proponen varias estrategias llegando a la conclusión de que es una cuestión sin resolver. Una buena técnica podría ser renumerar en primer lugar aquellos nodos con menor grado interno en cada bloque, pues un posible $fill - in$ en el bloque diagonal podría producir nuevos $fill - ins$ en las matrices subdiagonales.

6 Referencias

[1] P. R. ALMEIDA BENITEZ y J. R. FRANCO BRAÑAS, The Go-Away algorithm for Block Factorization of a Sparse Matrix. Course on algorithms for Sparse Large Scale Linear Algebraic Systems. NATO ASI SERIES. Ed. Kluwer. Londres. 1996. (To appear).

[2] E. CUTHILL, Several Strategies for Reducing the Bandwith of Matrices, Papers of the Symposium on Sparse Matrices and Their Applications, IBM Thomas J. Watson Research Center, New York, 1971.

[3] J.J. DONGARRA et al., Solving Linear Systems on Vector and Shared Memory Computers, SIAM, Philadelphia, 1991.

[4] I.S. DUFF, A.M. ERISMAN and J.K. REID, Direct Methods for Sparse Matrices, Monographs on Numerical Analysis, Oxford Press, Oxford, 1989.

[5] J.L. DE LA FUENTE O'CONNOR, Tecnologías Computacionales para Sistemas de Ecuaciones, Optimización Lineal y Entera, Ed. Reverté, Barcelona, 1993.

[6] K.A. GALLIVAN et al., Parallel Algorithms for Matrix Computations, SIAM, Philadelphia, 1991.

[7] A. GEORGE, Block Elimination of Finite Elements Systems of Equations, The IBM Symposia Series, pp. 101-114, Plenum Press, New York, 1972.

[8] A. GEORGE, An Automatic One-Way Dissection Algorithm for Irregular Finite Elements Problems, SIAM J. Num. Anal. 17, pp. 740-751, Philadelphia, 1980.

[9] A. GEORGE et al., Computer Solution of Large Sparse Positive Defined Systems, Prentice-Hall, London, 1981.

[10] A. GEORGE and E. NG, Waterloo Sparse Matrix Package. User guide for SPARSPAK-B. Research Report CS-84-37, Departament of Computer Science, University of Waterloo, Ontario, Canada, 1984.

[11] A. GEORGE and E. NG, An Implementation of Gaussian Elimination with Partial Pivoting for Sparse Systems, SIAM, J. Sci. Stat. Com. 6, pp. 390-409, Philadelphia, 1985.

[12] A. JENNINGS and J.J. MCKEOWN, Matrix Computation, Ed. John Wiley and Sons, Chichester (U.K.), 1992.

[13] D.J. ROSE, R.E. TARJAN and J.S. LUEKER, Algorithmic aspects of vertex elimination on graphs, SIAM, J. Comput., pp. 266-283, 1976.

[14] H.A.G. WIJSHOFF, Direct Methods for Solving Linear Systems, Papers of the Large-Scale Scientific Parallel Computing Course, Abingdon (U.K.), 1992.

[15] Z. ZLATEV, Computational Methods for General Sparse Matrices, Kluwer Academic Publishers, London, 1991.

Recibido: 30 Diciembre 1996